CS 225

#16: Binary Search Tree (BST)

February 20, 2018 · *Fagen-Ulmschneider, Zilles*

Traversal vs. Search:

- **Traversal** visits every node in the tree exactly once.
- **Search** finds one (or more) element(s) in the tree.

Breadth First Traversal + Search:

Depth First Traversal + Search:

Dictionary ADT

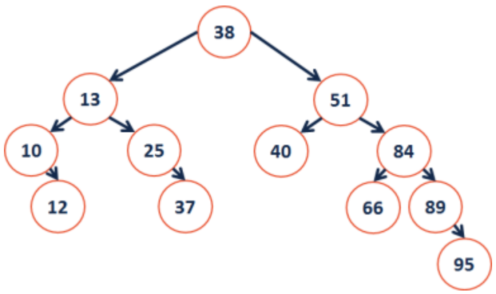
Dictionary.h

```

class Dictionary {
  public:






  private:


};
```

Runtime Analysis on a Binary Tree:

- Find an element: Best case? Worst case?
- Insertion of a sorted list of elements?
  Best case? Worst case?
- Running time bound by?

A Searchable Binary Tree?



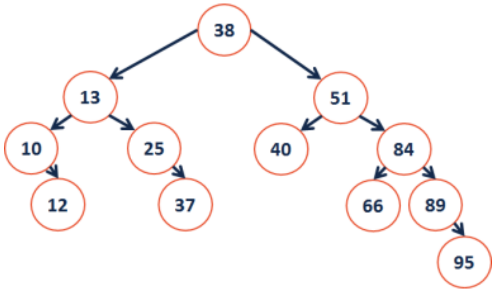
Binary Search Tree Property:

Finding an element in a BST:

BST.hpp

```
template <typename K, typename V>
___________________________________ find(const K & key)
const {






}

template <typename K, typename V>
_________________________________________________ _find
    (TreeNode *& root, const K & key) const {
















}
```

Inserting an element into a BST:

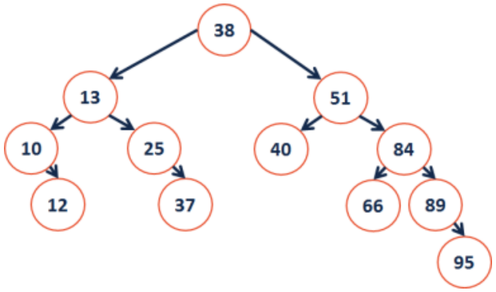


BST.hpp

```
template <typename K, typename V>
void BST<K, V>::_insert(TreeNode *& root, K key, V value)
{











}
```

Running time? ________________ **Bound by?** _______________

What if we did not pass a pointer by reference?



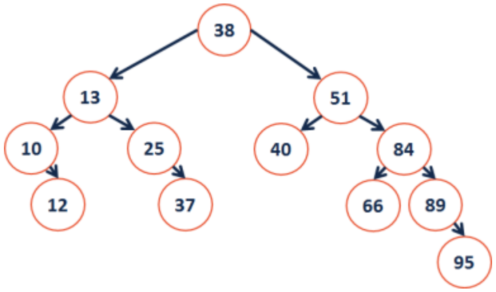
Removing an element from a BST:

`_remove(40)`

`_remove(25)`

`_remove(10)`

`_remove(13)`



One-child Remove	Two-child remove

BinaryTree.hpp

```
template <class K, class V>
void BST<K,V>::_remove(TreeNode *& root, const K & key) {






}
```

Running time? ________________ **Bound by?** _______________

CS 225 – Things To Be Doing:

1. Theory Exam 2 Topics List Posted (exam next week)
2. MP3 extra credit on-going; MP3 due Monday, Feb. 25
3. Upcoming Lab: lab_trees
4. Daily POTDs